CLAIMS

We claim:

1. A processing system comprising:

a processor that is configured to execute program instructions contained in a memory,

the processor including:

a program counter that is configured to contain a next-instruction-address, and

a stack that is configured to contain at least one return address corresponding to an execution of a subroutine call instruction,

the program instructions including:

a branch-else-return instruction that causes the processor to:

place a branch-address into the program counter as the next-instruction-address if an associated branch condition is in a first state, and,

place the at least one return-address into the program counter as the next-instruction-address if the associated branch condition is in a second state.

2. A processing system comprising:

a processor that is configured to execute program instructions that relate to data-items that occupy multiple words in a memory, the processor comprising:

a status register that includes status flags,

the status flags including at least one flag that is dependent upon corresponding multiple words of a select data-item.

3. The processing system of claim 2, wherein

the at least one flag includes at least one of:

a data-zero flag that indicates that each word of the multiple words forming the select data-item contains a zero value,

a data-one flag that indicates that each word of the multiple words forming the select data-item contains a zero value except a least-significant word of the multiple words forming the current data-item, and this least-significant word contains a value of one, and

a data-highest flag that identifies a most significant non-zero word of the multiple words forming the select data item.

4. A processing system comprising:

a processor that is configured to execute a current instruction from an instruction register, and

an operand register that is configured to provide an operand for processing by the processor in dependence upon the current instruction; and

wherein

the current instruction includes a constant-follows flag, and

the processor is configured to:

load a subsequent word into the operand register when the constant-follows flag of the current instruction contains a first

value, and

load the subsequent word into the instruction register at a next processor cycle when the constant-follows flag of the current instruction contains a second value.

5. A processing system comprising:

a processor that is configured to execute program instructions,

a memory that is configured to contain operands, each operand having a corresponding operand address in the memory and

at least one address register that is configured to contain an operand address; and

wherein

each of the at least one address registers is configured to:

receive the operand address from the processor, and

provide the operand address as an addressing input to the memory only.

6. The processing system as claimed in claim 5, wherein

the operand address lies within an operand-address-range, and

each of the at least one address registers is sized to be a minimum size required to contain a span of the operand-address-range.

7. The processing system of claim 5, wherein

at least one instruction of the program instructions effects a modification of at least two address registers upon execution of the at least one instruction.

8. The processing system of claim 5, wherein

the processor is further configured to provide an address-zero flag that is asserted when the operand address is zero,

the operand-address corresponds to a counting index, and

the at least one address register is further configured to decrement the operand address in response to a decrement command from the processor, thereby providing a counting operation based on the counting index.

9. The processing system as claimed in claim 5, wherein:

the operand address lies within an operand-address-range having a lower-address and an upper-address, and

the program instructions include at least one of:

a circular-increment instruction that

increments the operand-address in the at least one address register, and

resets the operand-address in the at least one address register to correspond to the lower-address when the operand-address in the at least one address register is greater than the upper-address,

a circular-decrement instruction that

decrements the operand-address in the at least one address register, and

resets the operand-address in the at least one address register to correspond to the upper-address when the operand-address in the at least one address register is less than

the lower-address,

thereby constraining the operand-address in the at least one address register to lie within the operand-address-range.

10. The processing system as claimed in claim 9, further including at least one condition flag that is associated with at least one of:

the operand-address equaling the lower-address, and

the operand-address equaling the upper-address.

11. A processing system comprising:

a processor that is configured to execute program instructions,

a memory that is configured to contain operands, each operand having a corresponding operand address in the memory,

wherein

the processor includes an arithmetic-unit, and

the arithmetic-unit is

operably coupled to the memory such that the arithmetic-unit receives a first operand from the memory only,

is further configured to receive a second operand from one only of the following: an output of the arithmetic-unit, and a constant, and

is further configured to produce the output based on at least one of the first operand and the second operand.

12. The processing system of claim 11, wherein

the arithmetic-unit includes only:

an adder having a first input and a second input that provides the output of the arithmetic-unit corresponding to an arithmetic sum of the first input and the second input;

a first-operand selector that is configured to form the first input as one of:

the first operand, and

a zero value;

a second-operand selector that is configured to form the second input as one of:

the second operand,

an inversion of the second operand,

a shift of the second operand, and

a zero value.

13. A processing system comprising:

a processor that is configured to execute program instructions,

each instruction of the program instructions being formatted in accordance with a format-type of a plurality of format types,

wherein

each format type of the plurality of format types comprises a plurality of fields that each facilitates an operation that is to be performed in parallel with the execution of each program instruction.

14. The processing system of claim 13, wherein

a substantial majority of the plurality of fields of at least one format type is common to a corresponding majority of the

plurality of fields of at least another format type.

15. The processing system of claim 13, wherein

a substantial majority of the plurality of fields in each format type corresponds to control elements of a microinstruction that controls the operation of switches and state devices within the processor.

16. The processing system of claim 13, wherein

the processor comprises:

a state machine that executes each instruction based on a microinstruction corresponding to each instruction,

a format mapper that is configured to:

associate each control field of each instruction to an associated control element in the microinstruction, in dependence upon the format type of the instruction, and

provide a default condition to other control elements of the microinstruction that are not associated with each instruction.

17. The processing system of claim 16, wherein

the default condition is also dependent upon at least one of:

the format type of the instruction, and

at least one control field of the instruction.

18. The processing system of claim 16, wherein

the default condition includes at least one of:

a load-zero condition that is configured to set at least one of the other control elements to a zero value,

a null condition that is configured to leave at least one of the other control elements unaffected,

a load-bit condition that is configured to set at least one of the other control elements to a value contained in the instruction, and

an increment condition that is configured to increment a value associated with at least one of the other control elements.

19. The processing system of claim 13, further comprising

a memory having an external data-in port and an external data-out port that are configured to facilitate a storage and retrieval of data-items to and from the memory, and

wherein

at least one field of the plurality of fields includes an address-select field,

the address-select field facilitates a selection of an external address-port that is configured to provide another processor direct access to locations in the memory that are addressed by the external address-port for storing and retrieving data items via the external data-in and data-out ports.

20. The processing system of claim 13, further comprising:

a plurality of storage elements, and

wherein

at least two fields of the plurality of fields are associated

with an identification of at least two storage elements of the plurality of storage elements, and

at least one instruction of the program instructions facilitates a parallel modification of the at least two storage elements upon execution of the at least one instruction.